



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER OF PATENTS AND TRADEMARKS
Washington, D.C. 20231
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/540,576	03/31/2000	Robert G. Field	SUN1P252/P4198	2536
22434	7590	04/10/2003		
BEYER WEAVER & THOMAS LLP P.O. BOX 778 BERKELEY, CA 94704-0778			EXAMINER	KISS, ERIC B
			ART UNIT	PAPER NUMBER
			2122	
DATE MAILED: 04/10/2003				

15

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)
	09/540,576	FIELD ET AL.
	Examiner Eric B. Kiss	Art Unit 2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 20 March 2003.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-19 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-19 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 30 October 2002 is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) The proposed drawing correction filed on _____ is: a) approved b) disapproved by the Examiner.
If approved, corrected drawings are required in reply to this Office action.
- 12) The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

- 13) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.
- 14) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).
- a) The translation of the foreign language provisional application has been received.
- 15) Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____
- 4) Interview Summary (PTO-413) Paper No(s) _____
- 5) Notice of Informal Patent Application (PTO-152)
- 6) Other: _____

DETAILED ACTION

1. The amendment of March 20, 2003, has been received and entered. Claims 1-19 are pending.

Response to Arguments

2. In response to Applicant's arguments regarding the rejection under 35 U.S.C. §102(b), based on public use or sale of the invention, this rejection has been more formally and explicitly expressed below.
3. Applicant's arguments filed on March 20, 2003, with regard to U.S. Patent No. 5,787,245 to You et al. have been fully considered but they are not persuasive.
 - a. Applicant argues in paragraph 2 of page 4:

It should be noted that there is a grave distinction between an object (an object of an object oriented programming language) and a formal specification which can be put into a code generator. Thus it is respectfully submitted that the Examiner, even in the broadest sense, cannot reasonably consider these to be the same.

However, Applicant acknowledges that the "TPrimitiveConnection" object disclosed by You et al. defines a protocol for communication between a client and a server (see the last sentence of paragraph 3 on page 4). In addition, it is submitted that an accepted definition of the term "specification" is: A detailed description of something (see p.325 of Microsoft Press

Computer User's Dictionary, 1998). It is further submitted that the "TPrimitiveConnection" provides a formal detailed description of the communication protocol between a client and a server. The TPrimitiveConnection class disclosed by You et al. is used within C++ source code to instantiate objects or define subclasses from which objects are instantiated. C++ is a well-known object-oriented programming language, which is implemented using a compiler comprising a code generator to process the source code to produce machine-executable computer program code. Therefore, the Examiner maintains that the TPrimitiveConnection disclosed by You et al. can be considered a formal specification which can be put into a code generator.

b. Applicant argues in paragraph 3 of page 4:

As such, the "T[P]r[i]mitiveConnection" object cannot be considered a specification in the context of the relevant arts. Moreover, the "T[P]rimitiveConnection" described in *You et al.* does not teach or suggest inputting a formal specification into a code generator.

The first portion of this argument has been addressed as set forth above. Furthermore, You et al. disclose TPrimitiveConnection as C++ source code defining an abstract base class (see column 52, line 30 through column 53, line 5). You et al. further disclose, "Communication between client and server are handled using TPrimitiveConnection objects" (see column 52, lines 8 and 9). This implies that the TPrimitiveConnection base class is used to generate TPrimitiveConnection objects. These objects are instances of the TPrimitiveConnection class or of a TPrimitiveConnection subclass (see column 52, lines 20-23). In either case, this inherently requires the parsing the TPrimitiveConnection class definition using a code generator in order to generate the corresponding code for the object instances of the class.

- c. Applicant further argues in paragraph 4 of page 4:

You et al. does not teach inputting a formal specification into a code generator, which in turn parses the formal specification to generate a front-end debugger and a back-end debugger such that the front-end debugger and back-end debugger are compatible with each other.

However, You et al. disclose that the TPrimitiveConnection base class is used to generate TPrimitiveConnection objects, as described above. These connection objects in turn provide a front-end debugger (client) portion and a back-end debugger (server) portion (see column 52, lines 8-19). You et al. further disclose the front-end debugger program (client debugger) and back-end debugger program (debugger server) being compatible with each other (see Fig. 2, in which the client debugger is shown as interfacing with the debugger server).

d. In light of the above arguments, the Examiner maintains that You et al. disclose inputting a formal specification into a code generator, which in turn parses the formal specification to generate a front-end debugger portion and back-end debugger portion, such that the front-end debugger program and the back-end debugger program are compatible with each other.

Moreover, the Examiner maintains that “TPrimitiveConnection”, as disclosed by You et al. provides the formal specification. See the 35 USC § 102 and 35 USC § 103 rejections that follow.

Claim Rejections - 35 USC § 112

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

5. Claims 6, 7, 10, 13, 14, 17, and 19 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claims 6, 7, 10, 13, 14, 17, and 19 contain the trademark/trade name JAVA. Where a trademark or trade name is used in a claim as a limitation to identify or describe a particular material or product, the claim does not comply with the requirements of 35 U.S.C. 112, second paragraph. See *Ex parte Simpson*, 218 USPQ 1020 (Bd. App. 1982). The claim scope is uncertain since the trademark or trade name cannot be used properly to identify any particular material or product. A trademark or trade name is used to identify a source of goods, and not the goods themselves. Thus, a trademark or trade name does not identify or describe the goods associated with the trademark or trade name. In the present case, the trademark JAVA is improperly relied upon in the claim to incorporate the technical features of a particular programming language environment. However, the trademark JAVA can only properly define the source of the programming language environment, namely Sun Microsystems, Inc. Accordingly, the identification/description is indefinite.

Claim Rejections - 35 USC § 102

6. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

Art Unit: 2122

7. Claims 1, 8, 12, 15, and 18 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,787,245 to You et al.

You et al. disclose inputting a formal specification (TPrimitiveConnection; see column 52, lines 8-27) into a code generator (client debugger object) which in turn parses the formal specification to generate a front-end debugger (client debugger object; see column 4, lines 28-37) portion (connection object; see column 63, lines 25-28) and a back-end debugger (server debugger object) portion (reverse connection object; see column 57, lines 35-41). A communication protocol is enabled between the front-end debugger (client debugger object) and the back-end debugger program (server debugger object), wherein the communication protocol is defined by the formal specification (TPrimitiveConnection). You et al. further disclose a computer readable medium including computer program code (see column 80, lines 33-65) and a computer system (see column 79, lines 13-55) for performing the aforementioned actions. You et al. further disclose the front-end debugger program (client debugger) and back-end debugger program (debugger server) being compatible with each other (see Fig. 2, in which the client debugger is shown as interfacing with the debugger server).

Claim Rejections - 35 USC § 103

8. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

9. Claims 2, 3, 9 and 13 are rejected under 35 U.S.C. 103(a) as being unpatentable over You et al. as applied to claims 1 and 12, respectively, above.

As per claims 2 and 3, although You et al. disclose with such a C++ object-oriented programming language implementation and fails to disclose a Java object-oriented programming language method, one having ordinary skill in the computer art would recognize that the You et al. system can be implemented using a wide number of known object-oriented programming languages, including the Java programming language. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to utilize Java programming language code running on a virtual machine to implement the method of You et al. One would be motivated to do so in order to gain the platform independence that the Java programming language provides.

As per claim 9, although You et al. fail to teach the use of a declarative language, one having ordinary skill in the computer art would recognize that a specification could be written in any programming language style, including such a known declarative language. One would be motivated to do so because a declarative language is rule-based and is best suited to implementing a specification that is also rule-based. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to write the formal specification of You et al. in a declarative language because it is best-suited for such a purpose.

As per claim 13, although You et al. disclose with such a C++ object-oriented programming language implementation and fails to disclose a Java object-oriented programming language method, one having ordinary skill in the computer art would recognize that the You et al. system can be implemented using a wide number of known object-oriented programming languages, including the Java programming language. One would be motivated to do so in order

Art Unit: 2122

to gain the platform independence that the Java programming language provides. Further, although You et al. fail to teach the use of a declarative specification language, one having ordinary skill in the computer art would recognize that a specification could be written in any programming language style, including such a known declarative language. One would be motivated to do so because a declarative language is rule-based and is best suited to implementing a specification that is also rule-based. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to write the formal specification of You et al. in a declarative language because it is best-suited for such a purpose and to utilize Java programming language code running on a virtual machine to implement the front-end of the You et al. method to gain platform independence.

10. Claims 4 and 5 are rejected under 35 U.S.C. 103(a) as being unpatentable over You et al. as applied to claim 1 above, and further in view of U.S. Patent No. 5,901,315 to Edwards.

You et al. fail to teach the back-end debugger program, a portion of which comprising C language code, directly controlling and communicating with a virtual machine. However, Edwards teaches a back-end debugger program (debug engine, DE, and BE) comprising C language code (see column 4, lines 35-38) that directly controls and communicates with a virtual machine (see Figure 3). One having ordinary skill in the computer art would recognize that a back-end debugger program could be written in any known programming language that allows an interface to be established between a debuggee program and a debugger front-end. Further, a virtual machine that is controlled by and communicates with the debugger back-end is commonly used when the application being debugged comprises Java language code. It would

have been obvious to one having ordinary skill in the computer art at the time the invention was made to implement the teachings of Edwards into the method of You et al. in order to get the advantage of being able to interface with and debug a Java language program. One would be motivated to do so for debugging an application comprising Java language code using a non-Java language user interface.

11. Claims 6, 10, and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over You as applied to claims 1, 8, and 13 above, and further in view of Field et al., "The New Java Platform Debugger Architecture," contained in Birds of a Feather, '98 JavaOne conference schedule (hereinafter Field et al.).

Although You et al. disclose with such a protocol defined by a TPrimitiveConnection class, one having ordinary skill in the computer art would recognize that any known communication protocol could be used to implement the You et al. method and system, including a Java Debug Wire Protocol as once taught by Field et al. as a communication protocol between a debugger and a debuggee. One would be motivated to use the Java Debug Wire Protocol because it allows for cross-platform remote debugging. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to incorporate the Java Debug Wire Protocol into the method and system of You et al. to perform cross-platform debugging.

12. Claims 7, 11, 16, and 17 are rejected under 35 U.S.C. 103(a) as being unpatentable over You et al. as applied to claims 6, 8, and 15 above.

As per claims 7, 11, and 16, although You et al. do not disclose a method of, or computer code for, generating HTML documentation of the protocol, one having ordinary skill in the computer art would recognize that the specific procedures and data packet formats necessary for sending and receiving data for a particular protocol are necessary in order to be able to implement such. One would be motivated to generate documentation of a communication protocol to provide human-readable protocol documentation information to software developers enabling them to implement the protocol. Further, HTML is a platform-independent document format, and one would be motivated to use HTML for the purpose of generating the documentation to allow it to be read on different platforms. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to incorporate the generation of HTML protocol documentation into the method and computer code of You et al. to allow software developers using various computer platforms to read and understand the proper procedures involved in implementing the protocol.

As per claim 17, see rationale provided in item 14 above.

13. Claims 1-19 are rejected under 35 U.S.C. 102(b) based upon a public use or sale of the invention or, in the alternative, under 35 U.S.C. 103(a) as obvious over the presentation given in a public forum on March 26, 1998 by the Applicant as evidenced by "JavaOne 1998 Presentation" (submitted in Information Disclosure Statement filed October 30, 2002 and hereinafter *Slideshow*), along with "Birds of a Feather, '98 JavaOne conference schedule", (cited in previous office action and hereinafter *Schedule*).

The cited presentation (given in a public forum on March 26, 1998; see Schedule, p. 34) disclosed:

a formal specification defining a communication protocol written in JAVA Debug Wire Protocol (JDWP) declarative specification language (see, for example, pages 1, 2, 4, and 13 of *Slideshow*);

a JAVA front-end debugger program portion running on a first virtual machine (see, for example, pages 2-4, 7, 8, and 10-14 of *Slideshow*);

the JAVA front-end debugger program portion comprising JAVA programming language code (see, for example, pages 2-4, 7, 8, and 10-14 of *Slideshow*);

a back-end JAVA debugger program portion controlling and communicating with a second virtual machine (see, for example, pages 2, 4-8, and 10-14 of *Slideshow*);

the back-end debugger program portion comprising C language code (see, for example, pages 4 and 5 of *Slideshow*); and

the JAVA front-end debugger program and JAVA back-end debugger program being compatible with each other (see, for example, pages 2, 4, 7, 8, and 10-14 of *Slideshow*).

Furthermore, the computer-readable medium and system of claims 15 and 18, respectively, are considered inherent in implementing the disclosed features described above.

It is unclear, based on materials made available to the Examiner, whether or not the cited presentation expressly disclosed inputting the formal specification into a code generator, parsing the formal specification, and generating the JAVA front-end debugger program portion and back-end JAVA debugger program portion from the formal specification after parsing.

However, the front-end and back-end debugger program portions are disclosed as based on an implementation of the JDWP specification. Official Notice is taken that in order to arrive at a machine-readable implementation of a human-readable specification, a compilation process comprising parsing the input specification and generating the output code has been well-known and commonly practiced in the computer art. An exemplary description of this practice can be found in Alfred V. Aho, et al., "Compilers, Principles, Techniques, and Tools," 1986, Addison-Wesley (hereinafter *Aho et al.*). For instance, Fig. 1.9 on page 10 of *Aho et al.* shows the phases of such a compilation process, including syntax analysis (or parsing) and code generation. The alternative to compiling is writing the code directly in assembly language or binary machine language, which is typically impractical. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to incorporate parsing and code generation into the disclosed implementation as a widely accepted means to achieve such an implementation.

It is further unclear, based on materials made available to the Examiner, whether or not the cited presentation expressly disclosed generating HTML code that contains a human-readable description of the protocol specification. However, Official Notice is taken that the specific procedures and data packet formats necessary for sending and receiving data for a particular protocol are necessary in order to be able to implement such. One would be motivated to generate documentation of a communication protocol to provide human-readable protocol documentation information to software developers enabling them to implement the protocol. Further, HTML is a platform-independent document format, and one would be motivated to use HTML for the purpose of generating the documentation to allow it to be read on different

platforms. Therefore, it would have been obvious to one having ordinary skill in the computer art at the time the invention was made to incorporate the generation of HTML protocol documentation into the method presented to allow software developers using various computer platforms to read and understand the proper procedures involved in implementing the protocol.

Conclusion

14. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Eric B. Kiss whose telephone number is (703) 305-7737. The examiner can normally be reached on Tue. - Fri., 7:30 am - 5:00 pm. The examiner can also be reached on alternate Mondays.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Gregory Morse can be reached on (703) 308-4789.

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks
Washington, DC 20231

Or faxed to:

(703) 746-7239 (for formal communications intended for entry)

Or:

(703) 746-7240 (for informal or draft communications, please label
'PROPOSED' or 'DRAFT')

Hand-delivered responses should be brought to Crystal Park II, 2121 Crystal Drive, Arlington, VA, 22202, Fourth Floor (Receptionist).

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

EBK/*EBK*
April 7, 2003

Gregory Morse
GREGORY MORSE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100